Problem 3: The Tree Movers

Given two binary search trees, A and B, with nodes identified by (that is, having keys equal to) positive, non-zero integers, and the use of commands "delete K" and "add K" (defined below), what is the smallest number of commands that can be used to transform tree A into tree B?

Recall that in a binary search tree, the keys of all nodes in the left subtree of a node with key K must be less than K. Similarly, the keys of all nodes in the right subtree of a node with key K must be greater than K. There are no duplicate nodes.

The "delete K" command will delete the tree (or subtree) with its root at the node with the key K. Deleting the root of the entire tree leaves an empty tree. The "add K" command will add a new node identified by the integer K. This node will naturally be a leaf node.

Since we seek to transform tree A into tree B, it follows that commands will be applied only to tree A; tree B is "read only."

It is easy to see that it should never require more than $N+1$ commands to achieve the transformation of A into B, since deletion of the root node of tree A followed by the addition of one node for each of the N nodes in B (in the proper order) will achieve the desired goal. Equally easy to determine is the minimum number of commands required: if A and B are identical, then zero commands are required.

Input

There will be multiple input cases. For each case, the input contains the description of tree A followed by the description of tree B. Each tree description consists of an integer N that specifies the number of nodes in the tree, following by the keys of the N nodes in an order such that N "add" commands would create the tree. The last case is followed by the integer –1. No node will have a key larger than 10^9, and N will be no larger than 100.

Output

For each case, display a single line containing the input case number (1, 2, …) and the number of commands required to transform tree A into tree B, formatted as shown in the examples below.

Sample Input	Expected Output
``4 5 2 7 4 6 5 3 7 1 4 9``	``Case 1: 5 commands.``
``0 0``	``Case 2: 0 commands.``
``1 100 0``	``Case 3: 1 command.``
``0 1 100``	``Case 4: 1 command.``
``3 100 49 37 2 200 152``	``Case 5: 3 commands.``
``−1``	